

B¹ 3. A method of providing software executable on a computer having a working memory, comprising:

providing demand-loadable components initially stored outside of said working memory, each component having an entry point comprising a constructor for an object, and
providing a Namespace in said working memory which provides in said working memory access to ones of said components as they become needed by applications running in said computer.

23. The system of Claim 24 wherein said storage memory comprises one of:

- (a) a memory within said computer;
- (b) a memory external of said computer.
- (c) the output of another software component such as a compiler.

B² 24. A system residing in a working memory of a computer and in a storage memory, said system comprising:

demand-loadable components initially stored in said storage memory, each component having an entry point comprising a constructor for an object, and
a Namespace in said working memory which provides in said working memory access to ones of said components as they become needed by applications running in said computer.

Add claims 33-37 as follows:

B³ 33. A method of operating a computer having a working memory wherein applications and objects may be loaded during run time, comprising:

- providing a Namespace in the computer;
- providing a kernel resident in the working memory at run time;
- providing a loadable interprocess communication manager resident at link time

outside of the working memory and dynamically loadable into the working memory at run time, the interprocess communication manager having an IUnknown pointer, a QueryInterface method and plural interfaces;

running an application in the computer;

the application presents to the Namespace the name of the interprocess communication manager;

in response to the Namespace being presented by the application with the name of the interprocess communication manager, the Namespace returning to the application the IUnknown pointer of the interprocess communication manager;

upon return of the IUnknown pointer, the application using the IUnknown pointer to call the QueryInterface method of the interprocess communication manager, and requesting through the QueryInterface method a pointer to a desired interface of the interprocess communication manager;

the QueryInterface method returning the desired interface, whereby the application can invoke a desired method through the interface.

34. The method of Claim 33, wherein the computer comprises a loader in the working memory, and wherein the Namespace, prior to providing the IUnknown pointer, performs the steps of:

determining whether the interprocess communication manager is currently registered in the Namespace;

if the interprocess communication manager is not currently registered, causing the loader to load the interprocess communication manager into the working memory and registering the interprocess communication manager in the Namespace.

35. A method of operating a computer having a working memory wherein applications and objects may be loaded during run time, the method comprising:

providing a Namespace in the computer;

providing a kernel resident in the working memory at run time;
providing a loadable virtual memory manager resident at link time outside of the memory and dynamically loadable into the working memory at run time upon demand of one of the application programs, the virtual memory manager having an IUknown pointer, a QueryInterface method and plural interfaces;

running an application in said computer;
said application presenting the name of the virtual memory manager to the Namespace;

in response to the Namespace being presented by the application with the name of the virtual memory manager, the Namespace returning to the application the IUknown pointer of the virtual memory manager;

upon return of the IUknown pointer, the application using the IUknown pointer to call the QueryInterface method of the virtual memory manager and to request through the QueryInterface method a pointer to a desired one of the plural interfaces of the virtual memory manager;

the QueryInterface method returning the desired interface, whereby the application can invoke a desired method through the Interface.

36. The method of Claim 35, wherein the computer comprises a loader in the working memory, and wherein the Namespace, prior to providing the IUknown pointer, performs the steps of:

determining whether the name of the virtual memory manager is currently registered in the Namespace;

if the name is not currently registered, causing the loader to load the virtual memory manager into the working memory and registering the virtual memory manager in the Namespace.

37. The method of Claim 35 wherein the plural interfaces of the virtual memory